

UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA



DESENVOLVIMENTO MOBILE – CRIAÇÃO DE UMA PLATAFORMA UWP DE APOIO AO PROCESSO DE INSPEÇÃO

Rui Manuel Marmeleira Poeiras

Mestrado em Engenharia Informática
Engenharia de Software

Versão Pública

Trabalho de Projeto orientado por:
Prof. Doutor Alcides Miguel Cachulo Aguiar Fonseca
e coorientado por:
Eduardo Miguel Neto Alves Silvério

2018

Agradecimentos

Em primeiro lugar, quero agradecer aos meus Pais, por todo o apoio que me deram, e continuam a dar, por nunca me deixarem desistir dos meus objetivos, mesmo quando os mesmos pareciam distantes, e por me terem ensinado a ser a pessoa que sou hoje. Estou eternamente grato. Obrigado MÃE. Obrigado PAI.

Quero agradecer à minha Avó, que tão importante foi na minha infância e criação, por me ter ensinado a ler e a fazer contas, e por ter ajudado a alimentar a criança traquinas, curiosa e sabichona que um dia fui. Sei que, apesar de não saberes, e de não o conseguires mostrar, estás muito orgulhosa de mim. Obrigado AVÓ.

Quero agradecer aos meus amigos, tanto aos de Vendas Novas, como os que criei ao longo da minha vida em Lisboa. Costuma-se dizer que amigos são a família que escolhemos, e esta família nunca me deixou ficar mal. Não vou escrever aqui todos os vossos nomes, mas estou-vos muito grato.

Quero agradecer aos meus colegas de faculdade que partilharam comigo tudo isto, em especial, ao Gustavo Correia, ao João Santos, e ao Pedro Caldeira, por nunca me terem deixado “perder o comboio”, e por me terem ajudado a perceber que era realmente isto que eu queria. Se não fossem vocês, certamente não tinha chegado até aqui.

Quero agradecer ao meu colega de estágio David Berto, foi um prazer partilhar esta experiência contigo.

Quero agradecer ao meu orientador da faculdade, o professor Alcides Fonseca, pela ajuda, paciência, e disponibilidade demonstrada.

Quero agradecer ao meu orientador da empresa, Eduardo Silvério, por todo o apoio que deu ao longo deste estágio, e por me “apadrinhar” nesta entrada no mercado de trabalho.

Por fim, quero agradecer a todo o pessoal da Accenture, que estive comigo neste estágio, e com quem tive oportunidade de trabalhar, em especial, ao Vitor Carabineiro, ao Filipe Cardoso, ao Diogo Antão, ao Pedro Couto, e ao Rui Sequeira. Todos vocês tiveram coisas para ensinar, e por isso, foram também meus “orientadores” ao longo destes nove meses.

Aos meus Pais, e à minha Avó.

Resumo

Com as tecnologias para desenvolvimento móvel em constante atualização, e a enorme gama de dispositivos existentes no mercado, torna-se imperativo que as organizações tenham sistemas de *software* como auxiliares para o trabalho feito no exterior.

O objetivo deste trabalho é desenvolver uma aplicação móvel que permita aos inspetores de uma determinada organização cliente realizar as suas atividades de inspeção no exterior, desmaterializando o processo. Para a realização de uma inspeção no exterior, o inspetor precisa de transportar uma grande quantidade de documentos e recolher bastante informação, que leva a que o processo seja moroso, não tirando, atualmente, qualquer partido das tecnologias existentes.

Este relatório descreve uma solução desenvolvida em *Universal Windows Platform* que vai de encontro às necessidades muito específicas do trabalho dos inspetores do cliente. Esta solução vai permitir aos utilizadores consultar e realizar as tarefas de inspeção no exterior, recorrendo apenas a um *tablet*. Para isto ser possível é necessário que a aplicação permita a consulta de documentos, a recolha de assinaturas e o preenchimento de formulários, de uma forma prática e intuitiva. Assim, para além de desmaterializar o processo, livrando os inspetores do uso de papel e caneta, a inserção dos dados recolhidos para o sistema do cliente também será agilizada pois estes dados serão comunicados diretamente do *tablet* para o sistema central.

Palavras-chave: *Desenvolvimento Móvel, Tablet, Inspeção, Universal Windows Platform, Desmaterialização*

Abstract

With constantly evolving mobile development technologies and the huge range of mobile devices on the market, it becomes imperative that organizations have software systems to aid in out-of-the-office work.

The objective of this work is to develop a mobile application that allows the inspectors of a given client organization to carry out their inspection activities on the go, dematerializing the process. To carry out an inspection, the inspector needs to carry several documents and gather a lot of information, which leads to a time-consuming process, not taking any advantage of the technologies.

This document describes a solution developed in Universal Windows Platform that meets the very specific needs of the customer inspectors work. This solution will allow users to access and perform the inspection tasks on the go, using only one tablet. For this to be possible, it is necessary that the application allows for the reading of documents, signing documents and form filling, in a practical and intuitive way. Thus, in addition to dematerializing the process, freeing the inspectors from the use of paper and pen, the insertion of collected data in the customer's system will also be streamlined, as this data will be communicated directly from the tablet to the central system.

Keywords: *Mobile Development, Tablet, Inspection, Universal Windows Platform, Dematerialization*

Conteúdo

Capítulo 1	Introdução.....	1
1.1	Contextualização	1
1.1.1	Instituição de Acolhimento	2
1.1.2	A Equipa.....	2
1.2	Motivação	2
1.3	Objetivos.....	3
1.4	Estrutura do documento.....	3
Capítulo 2	Estado da Arte	5
2.1	Trabalho Relacionado.....	5
2.2	Tecnologias Estudadas	7
2.2.1	Desenvolvimento Nativo de Aplicações <i>Mobile</i>	7
2.2.2	Desenvolvimento Multiplataforma de Aplicações <i>Mobile</i>	10
2.2.3	Tecnologias de Base de Dados para Aplicações <i>Mobile</i>	12
Capítulo 3	Metodologia e Planeamento	15
Capítulo 4	Levantamento de Requisitos	17
Capítulo 5	Arquitetura	19
Capítulo 6	Desenvolvimento.....	21
Capítulo 7	Considerações Finais.....	23
7.1	Trabalho realizado	23
7.2	Principais obstáculos	24
7.3	Reflexão critica.....	24
7.4	Trabalho futuro	25
Bibliografia		27

Lista de Figuras

Figura 1 - Stack de software iOS [12].....	8
Figura 2 - Stack de software Android [14]	9
Figura 3 - Universo UWP [15].....	9
Figura 4 - Arquitetura de uma aplicação Cordova [18]	11
Figura 5 - Xamarin vs. Xamarin Forms [20]	12

Lista de Tabelas

Tabela 1 - Comparação entre bases de dados mobile	14
---	----

Abreviaturas

ACID – *Atomicity, Consistency, Isolation, Durability*

AES – *Advanced Encryption Standard*

API – *Application Programming Interface*

BD – *Base de Dados*

BLOB – *Binary Large Object*

CSS – *Cascading Style Sheets*

DI – *Departamento de Informática*

DPEI – *Dissertação/Projeto de Engenharia Informática*

FCUL – *Faculdade de Ciências da Universidade de Lisboa*

GPS – *Global Positioning System*

HTML – *HyperText Markup Language*

HTTP – *HyperText Transfer Protocol*

HTTPS – *HyperText Transfer Protocol Secure*

IDE – *Integrated Development Environment*

JSON – *JavaScript Object Notation*

MEI – *Mestrado em Engenharia Informática*

MVC – *Model-View-Controller*

MVP – *Model-View-Presenter*

MVVM – *Model-View-ViewModel*

ORM – *Object-Relational Mapping*

PDF – *Portable Document Format*

REST – *Representational State Transfer*

SDK – *Software Development Kit*

SOAP – *Simple Object Access Protocol*

SQL – *Structured Query Language*

SSL – *Secure Sockets Layer*

TCP – *Transmission Control Protocol*
TLS – *Trasnport Layer Security*
UI – *User Interface*
URI – *Uniform Resource Identifier*
UWP – *Universal Windows Platform*
WS – *Web Service*
WSDL – *Web Services Description Language*
XAML – *eXtensible Application Markup Language*
XML – *eXtensible Markup Language*

Capítulo 1

Introdução

Este Relatório Final é realizado no contexto da disciplina de Dissertação/Projeto de Engenharia Informática (DPEI) do Mestrado em Engenharia Informática (MEI) da Faculdade de Ciências da Universidade de Lisboa (FCUL). O Projeto sobre o qual o relatório trata encontra-se inserido na área do Desenvolvimento *Mobile*, e foi realizado no contexto empresarial, num regime de estágio feito na empresa Accenture – Consultores de Gestão S.A. no período compreendido entre 2 de outubro de 2017 e 29 de junho de 2018.

Este capítulo contém uma contextualização do projeto, bem como os objetivos do mesmo e uma descrição da organização do relatório.

1.1 Contextualização

A área do desenvolvimento *mobile* e computação móvel é cada vez mais um tópico pilar da Engenharia de *Software*. Gradualmente os dispositivos *mobile*, como os *smartphones*, os *tablets* e os *wearables*, têm vindo a roubar terreno aos tradicionais *desktops*, sendo cada vez mais utilizados como ferramentas de trabalho e lazer. Um estudo realizado em 2016 pela *StatCounter* mostra que já existem mais dispositivos *mobile* a aceder à Internet do que *desktops* [1]. Apesar da diferença ainda ser pequena, a tendência é para que a utilização destes dispositivos seja cada vez maior.

Com os números da utilização de dispositivos *mobile* a aumentar cada vez mais, aumentam também o número de aplicações para os mesmos disponíveis no mercado. Segundo o portal de estatística *Statista* existem cerca de 6.4 milhões de aplicações *mobile* espalhadas entre *Google Play*, *Apple App Store* e *Windows Store* [2].

Este paradigma atual oferece novos desafios aos Engenheiros de *Software*. É necessário criar ferramentas aptas para um mundo portátil, de ligação à Internet sempre disponível, prontas para trabalhar num vasto número de dispositivos com *hardware* e

sistemas operativos diferentes. É preciso tirar ainda partido das capacidades multimodais dos dispositivos (*touch screen*, *stylus*, sensores) de forma a oferecer uma experiência agradável ao utilizador sem nunca perder o foco na funcionalidade.

É neste contexto atual de novos desafios e de necessidade de adaptação para os dispositivos *mobile* que se insere este projeto.

1.1.1 Instituição de Acolhimento

A instituição de acolhimento responsável pelo projeto que vai ser realizado é a Accenture – Consultores de Gestão S.A. referida daqui para a frente apenas como Accenture. A Accenture é uma empresa multinacional de consultoria de gestão que oferece serviços e soluções em diversas indústrias, sendo uma delas as Tecnologias da Informação. A organização da empresa está dividida em 5 áreas principais: estratégia, consultoria, digital, tecnologia e operações [3].

A sede portuguesa da Accenture situa-se na Avenida Eng. Duarte Pacheco, Amoreiras, Torre 1 piso 16, 1070-101, Lisboa.

O projeto é realizado nas instalações de um cliente da empresa de acolhimento.

1.1.2 A Equipa

Desde o primeiro dia de trabalho no cliente fui inserido numa equipa multidisciplinar composta por mais cinco elementos da instituição de acolhimento: dois *Senior Managers*, dois *Senior Analyst*, e um outro *Programmer* estagiário, na mesma situação que eu.

1.2 Motivação

Apesar do enorme desenvolvimento tecnológico que vivemos nos dias de hoje, existem ainda bastantes processos, sobretudo de cariz mais burocrático, que não tiram partido deste desenvolvimento. Existem de facto sistemas informáticos que dão apoio à maior parte destes processos. Ainda assim, há muito que pode ser feito de maneira a tirar ainda mais partido daquilo que os sistemas de informação modernos nos oferecem atualmente, e as novas tecnologias *mobile* devem passar por parte da solução.

O desenvolvimento deste estágio é feito para uma entidade cliente da empresa de acolhimento. Uma das responsabilidades da entidade cliente é a realização de inspeções, e todo o processo envolvente às mesmas, desde a planificação à execução. Este processo é, neste momento, realizado maioritariamente através da deslocação de um inspetor ao local da inspeção, fazendo um levantamento de dados através de

formulários/questionários, para que estes sejam posteriormente analisados e arquivados. Este processo de recolha de dados é feito com recurso à utilização de papel e caneta. Portanto, é do interesse do cliente arranjar uma solução que desmaterialize este processo tornando-o digital. A solução para este tipo de desmaterialização passa pela utilização de dispositivos *mobile*, nomeadamente o *tablet*, que devido as suas características como a portabilidade e a possibilidade de utilizar um *stylus* para escrever, o tornam um substituto perfeito para o uso de papel e caneta.

Com esta desmaterialização ganha-se tempo a transcrever os formulários e documentos para formato digital, permitindo também poupar papel e uniformizar as tarefas de inspeção.

O cliente pretende então a implementação de um *software* que possa ser utilizado num *tablet* para facilitar a execução de inspeções.

1.3 Objetivos

O objetivo do projeto realizado no âmbito do estágio, é a elaboração de uma aplicação *mobile* para ser utilizada em *tablets*, que auxilie o ato da inspeção no exterior. Pretende-se que esta solução incida sobre seis funcionalidades principais:

- O preenchimento de formulários eletrónicos durante os atos de inspeção, atualmente realizados em papel e caneta;
- A realização de inspeções, previamente planeadas e que incidem sobre um determinado sujeito alvo;
- A realização de ações de controlo, efetuadas a veículos de transporte;
- O levantamento de infrações;
- A assinatura digital de documentos, transversal a todas as ações acima referidas;
- A utilização de um mapa para orientação dos inspetores no decorrer dos atos de inspeção.

1.4 Estrutura do documento

Este documento está organizado da seguinte forma:

- No Capítulo 1, é feita uma introdução do projeto, falando do seu contexto, da instituição de acolhimento, dos objetivos e da motivação do mesmo;
- No Capítulo 2, é feita uma descrição sobre o estado da arte, nomeadamente, sobre as tecnologias exploradas, os conceitos relacionados com o projeto, e exemplos de aplicações de âmbito semelhante ao do projeto;
- No Capítulo 3, é explicada a metodologia usada durante a realização do projeto, são descritos alguns processos de engenharia de *software* utilizados, e é apresentado o planeamento do projeto;
- No Capítulo 4, são descritos os atores do sistema, as *user stories*, e os requisitos funcionais e não-funcionais do projeto;
- No Capítulo 5, é explicada a arquitetura do projeto, bem como, alguns conceitos sobre a mesma;
- No Capítulo 6, é descrito o desenvolvimento, e implementação do projeto, bem como, as principais dificuldades encontradas;
- No Capítulo 7, são elaboradas as considerações finais do projeto, alternativas que poderiam ter sido exploradas, e o trabalho futuro.

Capítulo 2

Estado da Arte

Este capítulo descreve o trabalho relacionado com o projeto em desenvolvimento, e apresenta uma amostra do universo de tecnologias existentes para desenvolvimento de aplicações móveis.

2.1 Trabalho Relacionado

Atualmente praticamente todas as organizações, sejam elas governamentais ou não, já possuem um sistema informático que automatiza e facilita algum trabalho das mesmas. Ainda assim, a grande maioria destes sistemas funciona apenas dentro de um escritório. O mais recente desafio prende-se com a disponibilização de sistemas informáticos para facilitar o trabalho no exterior, tirando partido de dispositivos móveis.

Quando procuram soluções para trabalho no exterior, existem sempre alguns requisitos intrínsecos que a organização pretende ver cumpridos: a capacidade de trabalhar tanto com uma ligação à internet como *offline*, a possibilidade de recolher vários tipos de dados tirando partido das funcionalidades do dispositivo, a facilidade de integrar esta componente com os sistemas de *software* que já existem internamente, e a desmaterialização do material usado no exterior, nomeadamente, o papel e a caneta [4].

Após alguma investigação, é possível verificar a existência de algumas aplicações no mercado que servem este propósito de recolher informações no exterior e automatizar processos de inspeção.

O *OSIMple* é uma aplicação que automatiza e desmaterializa o processo de inspeções no exterior, nomeadamente, inspeções a pontes [5]. Este processo, tal como o âmbito do projeto em análise, é altamente dependente de papel e caneta. Assim o *OSIMple* pretende simplificar o mesmo através de um sistema que se divide em duas partes. Primeiro, uma aplicação móvel que deve ser utilizada pelos inspetores, de modo a recolher os dados necessários, nomeadamente formulários, notas e fotografias do local

onde se realiza a inspeção. A segunda parte do sistema consiste numa plataforma *web* para onde toda a informação da inspeção é submetida de forma a ser posteriormente analisada, revista, e potencialmente editada. Apesar de ser um produto recente, o *OSIMple* é já utilizado em mais de quatrocentas pontes, e existe o interesse de alargar o âmbito do produto para outros tipos de inspeções no exterior [6].

O *iAuditor* é mais uma solução comercial que pretende simplificar o processo de inspeção baseado em *checklists*. Está disponível para Android, iOS e UWP, e permite a criação de formulários personalizados, sendo possível implementar lógica e fluxos específicos do trabalho que se se pretende fazer. Tal como o *OSIMple* combina uma componente móvel com uma componente *Web* de forma tornar mais fácil a análise dos dados recolhidos. É uma solução bastante popular sendo atualmente utilizada nas mais variadas indústrias, nomeadamente: construção, hotelaria, manufatura, vendas, transportes e logística [7]. Sendo assim, o *iAuditor* não pretende ser uma solução adaptada um determinado tipo de inspeção, mas sim generalizar os diversos processos de inspeção feitos com *checklists* em papel.

O *InspectTHIS!*, à semelhança do *iAuditor*, é uma solução generalizada para simplificar o ciclo de vida de uma inspeção nas mais diversas áreas de negócio. Nela é possível elaborar o planeamento da inspeção, a recolha dos dados durante a inspeção propriamente dita, e a análise posterior dos mesmos. Está também disponível para as três principais plataformas móveis (Android, iOS e UWP) [8].

Finalmente, o *Parsable*, é uma solução comercial que se foca, não no processo de inspeção, mas na desmaterialização do uso de papel dentro de grandes fábricas. O objetivo da mesma é a potenciação da Indústria 4.0 (a chamada quarta revolução industrial, que se foca na automação, *Internet of Things*, e troca de dados no interior das fábricas) [9]. Esta solução permite planear e publicar digitalmente documentos que descrevam processos complexos (de forma a uniformizar os mesmos), e ainda recolher e analisar dados sobre o que se está a passar dentro duma fábrica [10].

Apesar de, como se pode verificar, já existirem diversas soluções, tanto para facilitar o processo de inspeção no exterior, como para a desmaterialização de grandes quantidades de papel, há ainda um enorme défice de companhias que tirem partido deste tipo de *software*, continuando a realizar este tipo de trabalhos sem recurso a sistemas de informação. Esta resistência à mudança é explicada por vários fatores, nomeadamente o elevado preço das soluções existentes, e o facto das mesmas não serem, na maior parte das vezes, totalmente adaptadas aos problemas do cliente.

2.2 Tecnologias Estudadas

Esta secção descreve algumas tecnologias existentes para o desenvolvimento de aplicações móveis, nomeadamente tecnologias para desenvolvimento nativo, tecnologias para desenvolvimento multiplataforma e tecnologias de bases de dados *mobile*.

2.2.1 Desenvolvimento Nativo de Aplicações *Mobile*

iOS

É um sistema operativo para dispositivos móveis criado e desenvolvido pela empresa americana *Apple* e que corre exclusivamente em *hardware* da mesma (*iPhone*, *iPad* e *iPod Touch*). Foi lançado em 2007 como sistema operativo para o *iPhone* original e está atualmente na sua versão 11 lançada em 2017 [11]. As aplicações móveis para este sistema são distribuídas através da *Apple's App Store*, que contém cerca de 2.2 milhões de aplicações disponíveis para *download* [2].

O desenvolvimento de aplicações para *iOS* atualmente é feito através da linguagem de programação *Swift* (que sucede ao *Objective-C*), uma linguagem também ela criada e desenvolvida pela *Apple*.

A arquitetura do *iOS* está conceptualmente organizada em quatro camadas de *software* (tal como apresentado na Figura 1):

- *Cocoa Touch*, camada que contém as funcionalidades que são mais utilizadas pelos programadores de aplicações *iPhone*, nela estão contidas variadas bibliotecas de desenvolvimento, entre as quais, a responsável pelo UI da aplicação, pelos mapas, pelas notificações, pelas mensagens, entre outras.
- *Media Services*, camada que oferece capacidades de áudio, vídeo, animações e gráficos.
- *Core Services*, camada que oferece serviços fundamentais responsáveis pelo funcionamento das duas camadas acima, e que faz a ligação dos mesmos com o sistema operativo.
- *Core OS*, camada que ocupa a posição mais abaixo da *stack* de *software* e está imediatamente acima do *hardware* do dispositivo. Oferece serviços responsáveis pelo *networking*, interfaces de entrada e saída, e responsabilidades do sistema operativo como a gestão de memória, o sistema de ficheiros e *threads* [12].

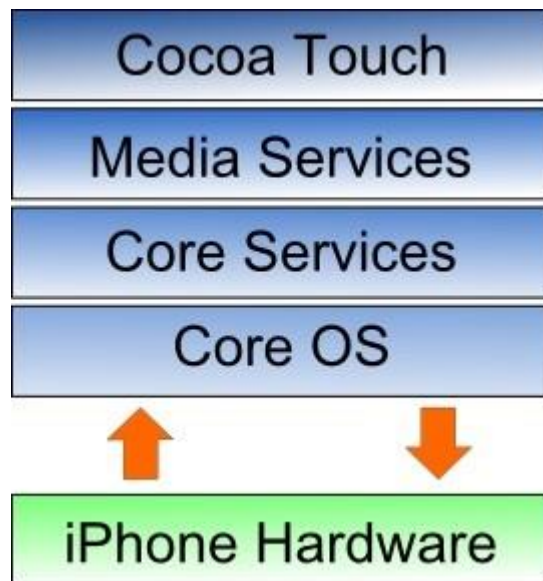


Figura 1 - Stack de software iOS [12]

Android

É um sistema operativo para dispositivos móveis *open source*, desenvolvido pela *Google*, baseado numa versão modificada do *kernel Linux*. Foi criado originalmente para ser utilizado em *tablets* e *smartphones*, e posteriormente para televisões, carros, *smart watches*, consolas, centros multimédia, câmaras digitais, entre outros. A primeira versão deste sistema operativo foi lançada em 2008 e a sua versão mais recente, a 8.1 *Oreo* foi lançada em 2017 [13]. É o sistema operativo móvel mais utilizado do mundo desde 2011, contando atualmente com mais de dois mil milhões de utilizadores, sendo que a sua loja de aplicações, a *Google Play Store*, conta com mais de três milhões e meio de aplicações disponíveis [2].

A *stack de software* do Android é tipicamente dividida em quatro camadas principais (tal como pode ser verificado na Figura 2):

- *Applications*, camada que representa as aplicações que estão instaladas no sistema Android, e que utilizam as APIs disponibilizadas pelas camadas abaixo.
- *Application Framework*, camada que expõe as APIs que são utilizadas para desenvolver aplicações Android (UI, comunicação, recursos, localização, entre outras).
- *Libraries* e *Runtime*, duas camadas que se encontram conceptualmente “lado a lado”. A camada de *Libraries* inclui bibliotecas nativas como: *WebKit*, *OpenGL*, *SQLite*, *Media*. A camada de *Runtime* é responsável pela DVM (*Dalvik Virtual Machine*), que funciona como uma JVM (*Java*

Virtual Machine) otimizada para dispositivos móveis. Ou seja, é a camada responsável por correr as aplicações Android.

- *Linux Kernel*, é a camada mais abaixo da arquitetura Android e é responsável pelos *drivers* do dispositivo, gestão de energia, gestão de memória e acesso de recursos [14].

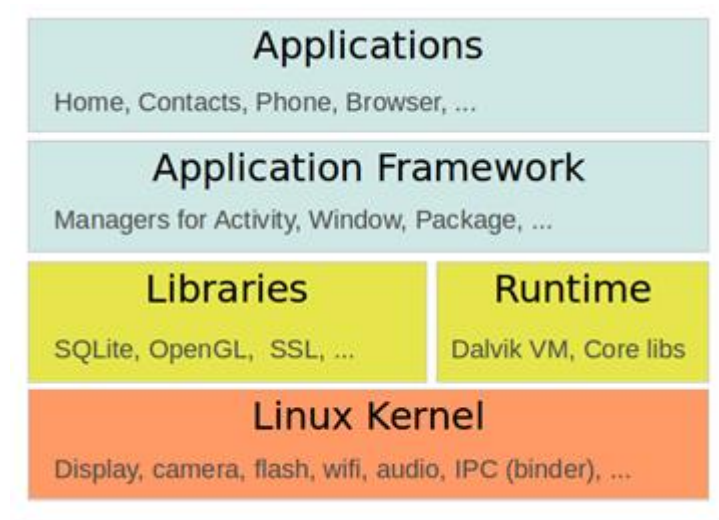


Figura 2 - Stack de software Android [14]

Universal Windows Platform

Universal Windows Platform (UWP) é uma plataforma de desenvolvimento de aplicações para o sistema operativo *Windows 10*. Fornece uma API única que permite desenvolver aplicações que corram em *Windows 10*, *Windows 10 Mobile*, *Xbox One* e *Hololens*, sem a necessidade de reescrever o código para cada uma das plataformas. O ecossistema do UWP pode ser visto na Figura 3.



Figura 3 - Universo UWP [15]

A ideia principal que sustenta esta API é a de que os utilizadores querem as suas aplicações prontas para utilizar em qualquer dispositivo *Windows*, independentemente da sua forma de *input* (*touch*, teclado, rato, comando ou *stylus*) ou do tamanho do seu ecrã. É implementada em C++ utilizando um subespaço do universo *.Net* (denominado *.Net Core*, uma versão mais pequena e revista do *.Net Framework*), e suporta desenvolvimento em C++, C#, VB.NET complementados com XAML.

Para além da API comum para as várias famílias de dispositivos que o UWP fornece, é possível utilizar APIs específicas de cada um dos dispositivos. Isto torna as aplicações desenvolvidas em UWP muito mais poderosas, pois estas podem verificar, em *runtime*, se uma dada API está presente antes de chamar a mesma, tirando assim partido de características importantes do dispositivo onde corre. Tem ainda a vantagem de utilizar *effective pixels* quando se faz o *front-end* da aplicação, de forma a que o *layout* responda de acordo com o número de pixéis disponíveis no ecrã.

Depois de desenvolvidas, as aplicações são distribuídas através de um *package .AppXm*, que fornece um mecanismo fidedigno de instalação através da *Windows Store*, permitindo o *deploy* e a atualização rápida das mesmas [15].

Outra particularidade das aplicações desenvolvidas em UWP, é que as mesmas executam apenas dentro do seu próprio contexto de segurança, ou seja, apenas podem utilizar as capacidades do dispositivo que são explicitamente declaradas no código fonte [16].

Para além disso, as aplicações UWP não conseguem aceder diretamente ao sistema de ficheiros, sendo apenas permitido ler e escrever na diretoria onde estão instaladas [17].

Os projetos em UWP são tipicamente desenhados e desenvolvidos utilizando o padrão *Model-view-viewmodel* que será explicado mais à frente.

2.2.2 Desenvolvimento Multiplataforma de Aplicações *Mobile*

Apache Cordova

O *Apache Cordova* é uma *framework* para desenvolvimento de aplicações móveis multiplataforma (Android, iOS e Windows UWP) criado originalmente pela *Nitobi*, e atualmente distribuído pela *Adobe Systems*. O *Cordova* permite aos programadores construir aplicações móveis com recurso a tecnologias *Web*, nomeadamente CSS3 e HTML5 para o *layout*, e JavaScript para a lógica [18]. As aplicações desenvolvidas nesta *framework* são híbridas, isto é, não são aplicações nativas (pois o seu UI é feito com recurso a uma *WebView*, em oposição às *frameworks* nativas), nem são aplicações

puramente *Web* (pois são distribuídas como aplicações e tem acesso a APIs nativas). A arquitetura de uma aplicação desenvolvida em *Cordova* pode ser vista na Figura 4.

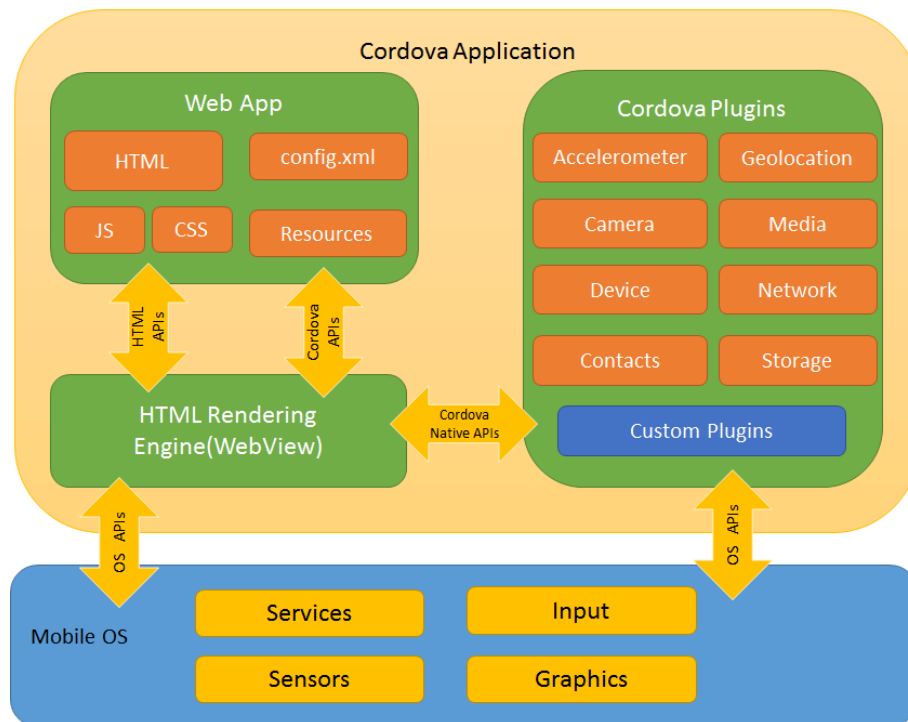


Figura 4 - Arquitetura de uma aplicação Cordova [18]

Appcelerator Titanium

O *Appcelerator Titanium* é uma *framework open-source*, para desenvolvimento de aplicações móveis multiplataforma (Android, iOS, Windows UWP) [19]. A principal diferença em relação ao *Cordova*, é que, o *Titanium* é uma *framework cross-platform*. Isto é, permite ao programador, desenvolver aplicações em JavaScript que são posteriormente interpretadas e convertidas em código nativo de cada plataforma. As aplicações *cross-platform* têm assim um desempenho superior em relação às híbridas, pois todo o processamento do UI é feito através de controlos nativos, ao invés de uma *WebView* ou *Browser*, melhorando assim a experiência de utilização.

Xamarin

O *Xamarin* é uma *framework* que permite desenvolver aplicações móveis multiplataforma (Android, iOS e Windows UWP) criada pela empresa homónima, e atualmente mantida e desenvolvida pela *Microsoft*. Sendo uma *framework cross-*

platform, permite com um código fonte partilhado (escrito utilizando a linguagem C#), criar aplicações nativas para cada uma das plataformas [20].

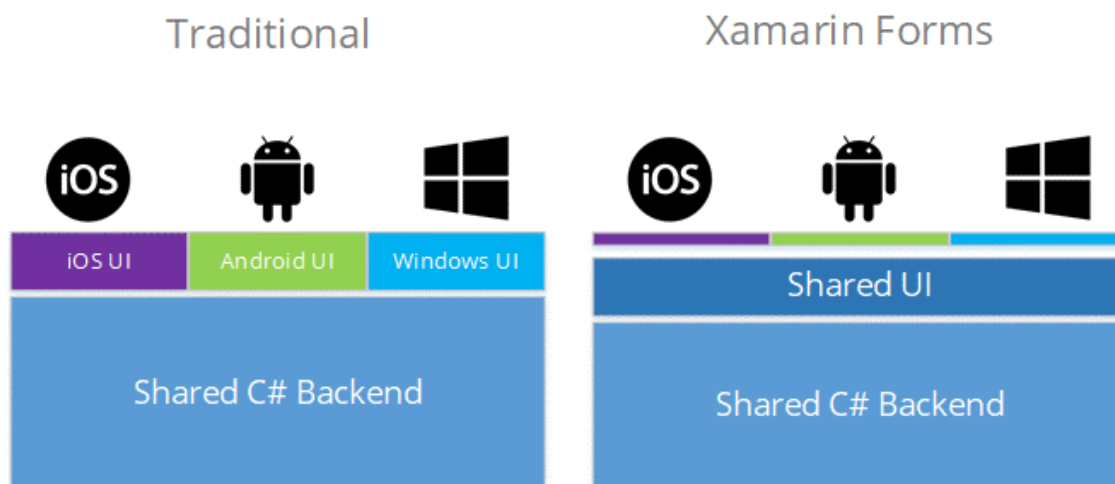


Figura 5 - Xamarin vs. Xamarin Forms [20]

O *Xamarin* fornece aos programadores duas formas diferentes de implementar os seus projetos: o *Xamarin* tradicional e o *Xamarin Forms*, tal como pode ser verificado na Figura 5. Estas duas abordagens oferecem diversas vantagens e desvantagens entre si.

No *Xamarin* tradicional, é possível partilhar código de *back-end* entre as três plataformas, mas é necessário implementar um projeto secundário para criar o UI de cada plataforma. Assim, o *Xamarin* tradicional é ideal para desenvolver projetos em que a *interface* seja complexa, que necessitem de muitas funcionalidades nativas, alta performance, e em que o tempo de desenvolvimento não seja um problema.

No *Xamarin Forms*, é possível com um único projeto, criar aplicações para as três plataformas, sem necessidade de código específico para cada, a não ser numa personalização muito específica. Assim, nesta modalidade, é oferecida ao programador uma camada de abstração adicional, que generaliza os controladores de UI de cada um dos sistemas. A grande desvantagem do *Xamarin Forms* é a dificuldade em dar às aplicações o *look-and-feel* específico de cada sistema e alguma perda de performance, mas em contrapartida, o tempo de desenvolvimento é consideravelmente menor e a partilha de código é quase total.

2.2.3 Tecnologias de Base de Dados para Aplicações *Mobile*

Estão disponíveis no mercado, várias variantes de bases de dados para serem implementadas em aplicações móveis. De uma maneira geral, existem uma série de características que as bases de dados para aplicações móveis devem cumprir [21]:

- devem ter um tamanho reduzido devido ao armazenamento interno destes dispositivos;
- devem ser *serverless*, ou seja, correr diretamente no dispositivo móvel;
- devem ter o mínimo de dependências possíveis;
- devem ser rápidas e seguras;
- devem ser fáceis de utilizar dentro do código fonte da aplicação;
- devem consumir pouca memória e pouca energia.

Segue em baixo uma tabela comparativa com cinco das mais utilizadas bibliotecas de bases de dados para aplicações móveis [22]:

Nome	Tipo	Descrição	Plataformas Suportadas
SQLite	Relacional	Biblioteca para bases de dados relacionais que apresenta uma implementação simples do SQL para dispositivos móveis.	Android iOS UWP
LevelDB	Chave-Valor	Biblioteca que implementa uma base de dados chave-valor, em que as chaves e os valores são armazenados como <i>array</i> de <i>bytes</i> , sendo os dados armazenados ordenados pela sua chave.	Android iOS UWP
UnQLite	Chave-Valor, Documentos	Biblioteca que expõe uma base de dados com duas interfaces, uma baseada em chave-valor e outra baseada em documentos.	Android, iOS
Berkeley DB	Relacional, Chave-Valor, Objetos, Documentos	Biblioteca que expõe uma base de dados com várias interfaces que permitem armazenar dados das mais diversas formas, tanto de forma relacional (SQL), como através de chave-valor, oferecendo ainda suporte para objetos Java e documentos XML.	Android iOS
Couchbase Lite	Documentos	Biblioteca que fornece uma base de dados baseada em documentos e em que os dados são armazenados	Android iOS

		recorrendo a ficheiros JSON.	
--	--	------------------------------	--

Tabela 1 - Comparação entre bases de dados mobile

SQLite

SQLite é uma biblioteca que implementa um sistema de gestão de bases de dados relacionais criada através da linguagem de programação C. A sua principal característica é o facto de ser *serverless*, ou seja, a base de dados está contida localmente no disco sendo acedida diretamente pela aplicação. Esta característica, aliada às suas poucas dependências e ao seu reduzido tamanho, tornam o *SQLite* uma tecnologia ideal para implementar uma base de dados local numa aplicação móvel. No *SQLite* cada base de dados é guardada num ficheiro separado, com extensão *.sqlite*. Este ficheiro pode ser armazenado tanto em memória, como no armazenamento interno do dispositivo.

Partilha características com outras tecnologias de base de dados relacionais como a conhecida sintaxe *SQL* e as transações *ACID* [23].

As transações *ACID* são uma característica muito importante das tecnologias de base de dados, pois estas garantem que cada operação lógica num determinado conjunto de dados respeita as seguintes propriedades [24]:

- *Atomicity*, garante que para qualquer transação o resultado será sempre “tudo ou nada”, ou seja, se uma parte da mesma falhar, então toda a transação falha, deixando a base de dados no estado que esta se encontrava antes da transação;
- *Consistency*, assegura que qualquer transação levará a base de dados de um estado válido para outro estado válido, isto é, quaisquer dados escritos numa base de dados devem ser válidos segundo as regras definidas na mesma;
- *Isolation*, confirma que a execução concorrente (em simultâneo) de transações resulta num estado igual ao que resultaria se cada transação fosse executada sequencialmente, isto é, controlo de concorrência;
- *Durability*, certifica que quando é feito *commit* a uma determinada transação esta será mantida permanentemente, mesmo em casos de eventuais perdas de energia ou *crash* do sistema.

Capítulo 3

Metodologia e Planeamento

Este capítulo encontra-se omissa por confidencialidade.

Capítulo 4

Levantamento de Requisitos

Este capítulo encontra-se omissa por confidencialidade.

Capítulo 5

Arquitetura

Este capítulo encontra-se omissa por confidencialidade.

Capítulo 6

Desenvolvimento

Este capítulo encontra-se omissa por confidencialidade.

Capítulo 7

Considerações Finais

Este capítulo contém uma visão geral sobre o decorrer do estágio realizado na empresa *Accenture*. Será então feita uma reflexão crítica do trabalho realizado, principais obstáculos encontrados e como foram ultrapassados, e ainda uma análise sobre o possível trabalho futuro.

7.1 Trabalho realizado

O trabalho descrito neste relatório foi realizado no âmbito do Projeto em Engenharia Informática, disciplina de conclusão do Mestrado em Engenharia Informática da FCUL. Foi realizado numa empresa de acolhimento, a *Accenture*, tendo ainda a particularidade de ter decorrido diretamente nas instalações de um cliente da mesma. Ao longo dos nove meses decorridos no estágio, foi possível pôr em prática diversos conhecimentos obtidos durante a licenciatura e mestrado, aplicando os mesmos no contexto de um projeto real.

O primeiro semestre foi dedicado à investigação de tecnologias que foram ou poderiam ter sido utilizadas, bem como, contextualização com o negócio do cliente e problemas que o mesmo queria ver resolvidos. Este estudo de tecnologias foi complementado com a elaboração de várias aplicações simples, que serviram de exemplos para posteriormente se desenvolver o protótipo final. Foi ainda feita uma primeira análise de requisitos funcionais e não funcionais junto do cliente, uma análise dos possíveis riscos inerentes ao projeto, e elaboração de protótipos de alto nível da interface da aplicação.

O segundo semestre foi sobretudo dedicado ao desenvolvimento do protótipo final, que representa uma aproximação funcional, com grande parte dos requisitos implementados, da solução que o cliente pretende, bem como, à preparação e escrita deste relatório final.

7.2 Principais obstáculos

Os principais obstáculos encontrados ao longo da realização deste projeto, estiveram sobretudo relacionados com a falta de experiência do estagiário e com as dificuldades encontradas em definir os requisitos funcionais da aplicação junto do cliente.

A falta de experiência, deu origem a que muitas vezes existisse algum atraso na realização das funcionalidades. Isto deve-se sobretudo ao anterior desconhecimento da maior parte das tecnologias utilizadas no desenvolvimento do projeto, obstáculo este que foi ultrapassado através da leitura exaustiva de documentação das mesmas, e a elaboração de aplicações exemplo, tal como referido anteriormente. Para além da familiarização com as tecnologias, surgiram ainda outros obstáculos de nível técnico, nomeadamente a aplicação do modelo MVVM, que apesar de ser revelado bastante complicado inicialmente, acabou por se revelar uma excelente maneira de organizar aplicações deste género. Foram ainda encontradas dificuldades ao nível de algumas limitações da *framework* UWP (aliada às bibliotecas *.NET Core*), pois esta é ainda bastante recente, sendo que as suas bibliotecas estão a sofrer constantes alterações e acréscimos.

Em relação as dificuldades encontradas na definição de requisitos, isto deve-se ao facto de a aplicação desenvolvida estar envolvida num projeto de desmaterialização de grandes dimensões, que ainda agora está a começar. Isto levou a que o desenvolvimento desta aplicação não fosse propriamente uma prioridade atual do cliente, pois existe ainda muito a ser feito a nível do sistema central para que esta possa começar a ser utilizada.

7.3 Reflexão crítica

Sendo o principal objetivo deste estágio, o desenvolvimento de um projeto que proporciona-se ao estagiário uma experiência rica, tanto a nível académico, como a nível profissional, pode-se dizer que este objetivo foi cumprido.

Por um lado, foi possível pôr em prática numerosas *skills* desenvolvidas ao longo do curso, aproveitando ainda para ganhar conhecimentos práticos em novas ferramentas e modelos de desenvolvimento, o que enriquece ainda mais a experiência. Do ponto de vista profissional, considera-se o estágio uma excelente forma de ganhar conhecimento sobre como funciona o mercado de trabalho, e como a Engenharia Informática é posta em prática do lado das empresas.

Analisando a experiência um pouco mais a fundo, pode-se dizer que a *framework* UWP utilizada neste projeto, apesar de não ser muito popular, é bastante poderosa e está em crescimento. A sua utilização levou a aprendizagem da linguagem C# e das bibliotecas *.NET*, servindo como ponto de entrada para o universo do desenvolvimento de *software* para máquinas *Windows*.

Apesar da escolha de máquinas *Windows* ter sido feita pelo cliente, o que deixava apenas duas opções de desenvolvimento, desenvolver nativamente para *Windows*, ou desenvolver uma aplicação híbrida, seria relevante discutir outras opções. A aplicação poderia ter sido feita de forma híbrida, nomeadamente através da tecnologia *Xamarin.Forms* que neste momento aparenta ser a opção mais fiável para este tipo de desenvolvimento. Esta hipótese foi posta de lado, por se considerar, que não se iria conseguir tirar todo o proveito das capacidades nativas dos aparelhos. Por outro lado, a escolha poderia ter recaído em dispositivos *Android* ou *iOS*, fazendo desenvolvimento nativo para os mesmos. Dada a variedade de dispositivos *Android* no mercado, e a vasta gama de preços dos mesmos, a escolha poderia ter sido desenvolver a solução utilizando *Android*. Esta escolha teria ainda a vantagem da utilização nativa da API *Google Maps*, que dada a enorme dependência da solução desenvolvida em mapas, poderia ser uma enorme evolução em relação à utilização dos mapas *Bing*. Apesar de tudo a escolha de máquinas *Windows* é compreensível, visto que os utilizadores que vão utilizar a aplicação, estão habituados a utilizar soluções *Windows* no seu trabalho. De referir ainda que a opção por estes dispositivos híbridos *Windows*, leva a que os mesmos, possam ser simultaneamente utilizados, tanto como dispositivos móveis no decorrer do trabalho exterior, mas também, como computadores no decorrer do trabalho no interior das instalações.

7.4 Trabalho futuro

Em relação ao trabalho futuro existem ainda várias alterações e melhorias que precisam de ser feitas, tanto a nível da aplicação, como do sistema central e *back-office* que a suporta.

Para começar, o próximo passo deste projeto seria a implementação dos requisitos funcionais que até à data de escrita deste relatório ainda não estão feitos. Nomeadamente a finalização do fluxo de trabalho da Tarefa PreenchimentoFormulários e da Tarefa ControloDeZona, que ainda não se encontram concluídos. Para a finalização destas funcionalidades, é necessário aguardar por algumas alterações do lado do *back-office* do cliente. É ainda necessário definir alguns pormenores em relação à autenticação dos utilizadores e segurança da mesma. Seria também necessário realizar

mais testes à aplicação, nomeadamente testes unitários e testes de aceitação junto dos utilizadores.

É ainda relevante mencionar que, sendo esta uma aplicação cujo objetivo é suportar ações de inspeção no exterior, esta pode com algumas alterações, ser adaptada a diferentes tipos de inspeção para além das elaboradas pelo cliente.

Bibliografia

- R. Simpson, “Mobile and tablet internet usage exceeds desktop for first time worldwide,” 1 Novembro 2016. [Online]. Available: <http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide>. [Acedido em 1 Novembro 2017].
- Statista, “Number of apps available in leading app stores as of 1st quarter 2018,” 2018. [Online]. Available: <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>. [Acedido em 1 Maio 2018].
- Accenture, “Sobre a Accenture,” [Online]. Available: <https://www.accenture.com/pt-pt/company>. [Acedido em 1 Novembro 2017].
- A. Neagu, “Field Workforce Automation On Tablets: Enabling The Other Mobile Workforce,” Formotus, [Online]. Available: <https://www.formotus.com/blog/field-workforce-automation-on-tablets>. [Acedido em 4 Novembro 2017].
- OSIMple, “Solution,” [Online]. Available: <https://www.osimple.co/solution>. [Acedido em 2 Março 2018].
- A. Chan e M. Miller, “Launch HN: OSIMple (YC W18) – Automating data entry for inspectors,” [Online]. Available: <https://news.ycombinator.com/item?id=16485095>. [Acedido em 2 Março 2018].
- iAuditor, “Build checklists, conduct inspections, file reports,” [Online]. Available: <https://safetyculture.com/iauditor/>. [Acedido em 3 Março 2018].
- InspectTHIS!, “Mobile Inspection App,” [Online]. Available: <https://inspectthis.net/>. [Acedido em 3 Março 2018].
- M. Moore, “What is Industry 4.0? Everything you need to know,” Techradar, 24 Abril 2018. [Online]. Available: <https://www.techradar.com/news/what-is->

industry-40-everything-you-need-to-know. [Acedido em 25 Abril 2018].

Pasable, “Jobs Done Right, Every Time,” [Online]. Available:
[10] <https://www.parsable.com/>. [Acedido em 25 Abril 2018].

Wikipedia, “iOS,” [Online]. Available: <https://en.wikipedia.org/wiki/IOS>.
[11] [Acedido em 9 Fevereiro 2018].

Techotopia, “iPhone iOS 5 Architecture and SDK Frameworks,” [Online].
[12] Available:
https://www.techotopia.com/index.php/IPhone_iOS_5_Architecture_and_SDK_Frameworks. [Acedido em 10 Fevereiro 2018].

Wikipedia, “Android (operating system),” [Online]. Available:
[13] [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)). [Acedido em 10 Fevereiro 2018].

Techotopia, “An Overview of the Android Architecture,” [Online]. Available:
[14] https://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture. [Acedido em 10 Fevereiro 2018].

Microsoft, “What's a Universal Windows Platform (UWP) app?,” 7 Maio
[15] 2018. [Online]. Available: <https://docs.microsoft.com/en-us/windows/uwp/get-started/universal-application-platform-guide>. [Acedido em 19 Maio 2018].

Microsoft, “App capability declarations,” 16 Maio 2018. [Online]. Available:
[16] <https://docs.microsoft.com/en-us/windows/uwp/packaging/app-capability-declarations>. [Acedido em 19 Maio 2018].

Microsoft, “File access and permissions (Windows Runtime apps),” 13
[17] Outubro 2015. [Online]. Available: [https://docs.microsoft.com/en-us/previous-versions/windows/apps/hh967755\(v=win.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/apps/hh967755(v=win.10)). [Acedido em 19 Maio 2018].

Cordova, “Overview,” [Online]. Available:
[18] <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>. [Acedido em 24 Fevereiro 2018].

Appcelerator, “Native apps. Mobile APIs. Real-time analytics. One
[19] Platform,” [Online]. Available: <https://www.appcelerator.com/mobile-app-development-products/>. [Acedido em 24 Fevereiro 2018].

Xamarin, “Everything you need to deliver,” [Online]. Available:
[20] <https://www.xamarin.com/>. [Acedido em 4 Novembro 2017].

A. Parihar, “Five of the Most Popular Databases for Mobile Apps,” Trigent,
[21] 25 Abril 2017. [Online]. Available: <https://blog.trigent.com/five-of-the-most-popular-databases-for-mobile-apps/>. [Acedido em 24 Fevereiro 2018].

K. Roukounaki, “Five popular databases for mobile,” Developer Economics,
[22] 10 Setembro 2014. [Online]. Available: <https://www.developereconomics.com/five-popular-databases-for-mobile>.
[Acedido em 24 Fevereiro 2018].

SQLite, “About SQLite,” [Online]. Available:
[23] <https://www.sqlite.org/about.html>. [Acedido em 24 Fevereiro 2018].

Wikipedia, “ACID,” [Online]. Available: <https://en.wikipedia.org/wiki/ACID>.
[24] [Acedido em 25 Fevereiro 2018].

Wikipedia, “Software prototyping,” [Online]. Available:
[25] https://en.wikipedia.org/wiki/Software_prototyping. [Acedido em 30 Março 2018].

N. Lavanya e T. Malarvizhi, “Risk analysis and management: a vital key to
[26] effective project management,” Project Management Institute, 3 Março 2008.
[Online]. Available: <https://www.pmi.org/learning/library/risk-analysis-project-management-7070>. [Acedido em 10 Março 2018].

P. Papajorgji e P. Pardalos, “Use Cases and Actors,” em *Software Engineering*
[27] *Techniques Applied to Agricultural Systems*, Springer, Boston, MA, 2006, pp. 61,
62.

Mountain Goat Software, “User Stories,” [Online]. Available:
[28] <https://www.mountaingoatsoftware.com/agile/user-stories>. [Acedido em 17 Março 2018].

O. E. Gabry, “Requirements Engineering—Introduction (Part 1),”
[29] OmarElGabry's Blog, 13 Setembro 2016. [Online]. Available:
<https://medium.com/omarelgabrys-blog/requirements-engineering-introduction-part-1-6d49001526d3>. [Acedido em 4 Março 2017].

Sketch, “Sketch. The digital design toolkit,” [Online]. Available:
[30] <https://www.sketchapp.com/>. [Acedido em 31 Março 2018].

Zeplin, “Features - Developers,” [Online]. Available:
[31] <https://zeplin.io/features#developer>. [Acedido em 31 Março 2018].

I. Manolov, “Model-View-ViewModel (MVVM) Applications: General

[32] Introduction,” Microsoft Developer, [Online]. Available: https://blogs.msdn.microsoft.com/ivo_manolov/2012/03/17/model-view-viewmodel-mvvm-applications-general-introduction/. [Acedido em 14 Abril 2018].

Microsoft, “The MVVM Pattern,” 10 Março 2012. [Online]. Available: [33] [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/hh848246\(v=pandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/hh848246(v=pandp.10)). [Acedido em 8 Dezembro 2017].

Microsoft, “Data binding in depth,” 8 Fevereiro 2018. [Online]. Available: [34] <https://docs.microsoft.com/en-us/windows/uwp/data-binding/data-binding-in-depth>. [Acedido em 25 Abril 2018].

M. Fowler, “Inversion of Control Containers and the Dependency Injection pattern,” 23 Janeiro 2004. [Online]. Available: [35] <https://www.martinfowler.com/articles/injection.html>. [Acedido em 25 Abril 2018].

Wikipedia, “Hypertext Transfer Protocol,” [Online]. Available: [36] https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol. [Acedido em 4 Maio 2018].

Wikipedia, “HTTPS,” [Online]. Available: [37] <https://en.wikipedia.org/wiki/HTTPS>. [Acedido em 4 Maio 2018].

Wikipedia, “Representational state transfer,” [Online]. Available: [38] https://en.wikipedia.org/wiki/Representational_state_transfer. [Acedido em 4 Maio 2018].

Wikipedia, “SOAP,” [Online]. Available: [39] <https://en.wikipedia.org/wiki/SOAP>. [Acedido em 6 Maio 2018].

M. Pagani, “Desktop Bridge – The Migrate phase: invoking a Win32 process from a UWP app,” 19 Dezembro 2016. [Online]. Available: [40] <https://blogs.msdn.microsoft.com/appconsult/2016/12/19/desktop-bridge-the-migrate-phase-invoking-a-win32-process-from-a-uwp-app/>. [Acedido em 15 Junho 2018].

Microsoft, “Desktop Bridge,” 14 Maio 2018. [Online]. Available: [41] <https://docs.microsoft.com/en-us/windows/uwp/porting/desktop-to-uwp-root>. [Acedido em 19 Maio 2018].

Wikipedia, “Hamburger button,” [Online]. Available:

[42] https://en.wikipedia.org/wiki/Hamburger_button. [Acedido em 1 Junho 2018].

Microsoft, “MapControl Class,” [Online]. Available:
[43] <https://docs.microsoft.com/en-us/uwp/api/windows.ui.xaml.controls.maps.mapcontrol>. [Acedido em 1 Dezembro 2017].

Microsoft, “Web view,” [Online]. Available: <https://docs.microsoft.com/en-us/uwp/api/windows.ui.xaml.controls.maps.mapcontrol>. [44] [Acedido em 1 Dezembro 2017].

Microsoft, “Perform geocoding and reverse geocoding,” [Online]. Available:
[45] <https://docs.microsoft.com/en-us/windows/uwp/maps-and-location/geocoding>. [Acedido em 2 Dezembro 2017].

Microsoft, “NetworkInformation Class,” [Online]. Available:
[46] <https://docs.microsoft.com/en-us/uwp/api/windows.networking.connectivity.networkinformation>. [Acedido em 8 Junho 2018].

Microsoft, “Create and register an out-of-process background task,” 2 Agosto
[47] 2017. [Online]. Available: <https://docs.microsoft.com/en-us/windows/uwp/launch-resume/create-and-register-a-background-task>. [Acedido em 8 Junho 2018].

Microsoft, “Get the user's location,” 28 Novembro 2017. [Online]. Available:
[48] <https://docs.microsoft.com/en-us/windows/uwp/maps-and-location/get-location>. [Acedido em 8 Dezembro 2018].

Microsoft, “Windows.Data.Pdf Namespace,” [Online]. Available:
[49] <https://docs.microsoft.com/en-us/uwp/api/windows.data.pdf>. [Acedido em 1 Junho 2018].

Microsoft, “Pen interactions and Windows Ink in UWP apps,” 8 Fevereiro
[50] 2017. [Online]. Available: <https://docs.microsoft.com/en-us/windows/uwp/design/input/pen-and-stylus-interactions>. [Acedido em 9 Dezembro 2017].

Microsoft, “Using JavaScript Object Notation (JSON) (Windows Runtime app
[51] using C++, C#, or Visual Basic),” 13 Outubro 2015. [Online]. Available: [https://docs.microsoft.com/en-us/previous-versions/windows/apps/hh770289\(v=win.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/apps/hh770289(v=win.10)). [Acedido em 17 Junho 2018].

Microsoft, “Design and code UWP apps,” [Online]. Available:

[52] <https://developer.microsoft.com/en-us/windows/apps/design>. [Acedido em 16 Junho 2018].

Microsoft, “Split view control,” 5 Maio 2017. [Online]. Available:
[53] <https://docs.microsoft.com/en-us/windows/uwp/design/controls-and-patterns/split-view>. [Acedido em 17 Junho 2018].

SQLCipher, “Documentation,” [Online]. Available:
[54] <https://www.zetetic.net/sqlcipher/design/>. [Acedido em 2 Fevereiro 2018].

C# in Depht, “Implementing the Singleton Pattern in C#,” [Online].
[55] Available: <http://csharpindepth.com/Articles/General/Singleton.aspx>. [Acedido em 1 Março 2018].

H. Hakeem, “Android by example : MVVM +Data Binding -> Introduction
[56] (Part 1),” Medium, 10 Setembro 2017. [Online]. Available:
<https://medium.com/@husayn.hakeem/android-by-example-mvvm-data-binding-introduction-part-1-6a7a5f388bf7>. [Acedido em 14 Junho 2018].

A. Sinhal, “MVC, MVP and MVVM Design Pattern,” Medium, 3 Janeiro
[57] 2017. [Online]. Available: <https://medium.com/@ankit.sinhal/mvc-mvp-and-mvvm-design-pattern-6e169567bbad>. [Acedido em 14 Junho 2018].

Microsoft, “Understanding WS-Security,” [Online]. Available:
[58] <https://msdn.microsoft.com/en-us/library/ms977327.aspx>. [Acedido em 8 Junho 2018].